# TRSDOS Commands and Utilities

**APPEND** Adds one disk file onto the end of another.
```
APPEND FTW/TXT NORTX/TXT
```

**ATTRIB** Changes protection of specified file. (I or V, ACC, UPD, PROT)
```
ATTRIB OLD/DAT (I,ACC=JUL14,UPD=MOUSE,
PROT=READ)
```

**AUTO *command*** Automatically executes the specified TRSDOS command each time TRSDOS starts up. (AUTO by itself erases the automatic command.)
```
AUTO CLOCK      AUTO BASIC      AUTO
```

**BACKUP** Duplicates a system or data diskette.
```
BACKUP      BACKUP :0 :1
```

**BUILD** Creates an automatic command input file.
```
BUILD JOBFILE
```

**BASIC** Loads Disk BASIC interpreter. BASIC ' allows recovery of the program that was in memory before the return to TRSDOS.
```
BASIC      BASIC *
```

**CLEAR** Clears user memory and set top memory address.
```
CLEAR (START=9000,END=AA00,MEM=7000)
CLEAR
```

**CLOCK** Turns real-time clock display on/off.
```
CLOCK (ON)      CLOCK      CLOCK (OFF)
```

**CLS** Clears the screen.
```
CLS
```

**CONVERT** Model I to Model III program/data file conversion.
```
CONVERT
```

**COPY *oldfile newfile*** Copies a file.
```
COPY  FILE1/BAS  UPDFL/BAS
COPY  FILE/A  FILE/A:1    COPY  FILEA/BAS:0 :1
```

**CREATE *filename*(LRL = *aaa*, REC = *bbb*)** Creates a preallocated file.
```
CREATE JOBFILE (LRL=256, REC=50)
```

**DATE *newdate*** Sets or displays the current date.
```
DATE 07/10/80      DATE
```

**DEBUG** Starts debug monitor.
```
DEBUG (turns monitor ON) Q (turns monitor OFF)
```

**DIR :*d*(INV, SYS, PRT)** Lists the diskette directory (INVisible or SYStem) on drive d on the Display or Printer (PRT).
```
DIR :1 (INV)      DIR :0 (PRT)
```

**DO *command-line*** Begin auto command input from disk file.
```
DO BEGIN
```

**DUAL (*switch*)** Duplicates output to video and printer.
```
DUAL (ON)      DUAL      DUAL (OFF)
```

**DUMP *file*** Dumps content of RAM into a machine-language program disk file. (START = *aaaa*,END = *bbbb*,TRA = *cccc*,RELO = *dddd*)
```
DUMP DATA/CIM:1 (START=0000,END=B05F)
```

**ERROR *number*** Displays an error message.
```
ERROR 8?
```

**FORMS (WIDTH = *aaa*,LINES = *bbb*)** Set printer parameters.
```
FORMS (WIDTH=80,LENGTH=55)
```

**FORMAT** Initializes a diskette into tracks and sectors.
```
FORMAT :1      FORMAT
```

**FREE** Lists a diskette's allocation map to the Display or Printer (PRT).
```
FREE :1      FREE :0 (PRT)
```

**HELP *command*** Explanation of TRSDOS command.
```
HELP BACKUP
```

**KILL *file*/EXT:*d*** Deletes a file from directory; frees space allocated to that file.
```
KILL FL/BAS:1      KILL /CMD:0
```

**LIB** Lists library commands.
```
LIB
```

**LIST *file* (PRT, SLOW, ASCII)** Lists contents of a file to the Display or Printer.
```
LIST PROG1/TXT (PRT)      LIST JOBFILE/BLD
                          (ASCII)
```

**LOAD *file*** Loads a machine-language file into memory.
```
LOAD GRAPHICS
```

**LPC** Special printer driver for some printers.
```
LPC
```

**MASTER (DRIVE = *a*)** Forces a drive to be the Master Read/Write drive. MASTER releases any drive defined as Master Drive.
```
MASTER (DRIVE=1)      MASTER
```

**MEMTEST** Test memory (ROM and RAM).
```
MEMTEST
```

**PATCH *file* (ADD = *aaaa*,FIND = *bb*,CHG = *cc*)** Change the contents of a disk file.
```
PATCH JOBFILE/BLD (ADD=5280,FIND=CD2C25,
CHG=C32C27)
```

**PAUSE *message*** Pauses for operator action or message.
```
PAUSE  INSERT DISKETTE *2
```

**PROT :*d* (PW, LOCK)** Changes file and diskette passwords.
```
PROT :1 (PW, LOCK)
```

**PURGE :*d* (*file-type*)** Deletes files. (SYS, DATA, ALL, INV).
```
PURGE :1 (INV)      PURGE :0
```

**RELO *file* (ADD = *aaaa*)** Changes location where program loads into memory.
```
RELO JOBFILE/BLD (ADD=6578)
```

**RENAME *file* TO *file*** Renames a file.
```
RENAME MRS/BAS TO MS/BAS
```

**ROUTE (SOURCE = *aa*,DESTIN = *bb*)** Routes I/O devices.
```
ROUTE (SOURCE=PR, DESTIN=DO)
```

**SETCOM (OFF,WORD = *a*,BAUD = *bbb*,STOP = *c*, PARITY = *d*, *mode*)** Sets up RS-232C communications or display status.
```
SETCOM (WORD=7,BAUD=300,STOP=1,
PARITY=0,WAIT)      SETCOM
```

**TAPE (S = *a*,D = *b*)** Executes tape transfer operator.
```
TAPE (S=D,D=T)
```

**TIME *hh:mm:ss*** Resets or gets the time.
```
TIME 14:12:30      TIME
```

**WP (DRIVE = *a*)** Write-protects a diskette.
```
WP (DRIVE=1)      WP
```

**XFERSYS** Transfers system files.
```
XFERSYS
```

# TRSDOS Error Messages

| | |
|---|---|
| 0 | No Error Found |
| 1 | CRC Error During Disk I/O |
| 2 | Disk Drive Not In System |
| 3 | Lost Data During Disk I/O |
| 4 | CRC Error During Disk I/O |
| 5 | Disk Sector Not Found |
| 6 | Disk Drive Hardware Fault |
| 7 | "Undefined Error Code" |
| 8 | Disk Drive Not Ready |
| 9 | Illegal I/O Attempt |
| 10 | Required Command Parameter Not Found |
| 11 | Illegal Command Parameter |
| 12 | Time Out On Disk Drive |
| 13 | I/O Attempt To Non System Disk |
| 14 | Write Fault On Disk I/O |
| 15 | Write Protected Disk |
| 16 | Illegal Logical File Number |
| 17 | Directory Read Error |
| 18 | Directory Write Error |
| 19 | Invalid File Name |
| 20 | GAT Read Error |
| 21 | GAT Write Error |
| 22 | HIT Read Error |
| 23 | HIT Write Error |
| 24 | File Not Found |
| 25 | File Access Denied Due To Password Protection |
| 26 | Directory Space Full |
| 27 | Disk Space Full |
| 28 | Attempt To Read Past EOF |
| 29 | Attempt to Read Outside of File Limits |
| 30 | No More Extents Available |
| 31 | Program Not Found |
| 32 | Invalid Drive Number |
| 34 | Attempt To Use Non Program File As a Program |
| 35 | Memory Fault During Program Load |
| 36 | "Undefined Error Code" |
| 37 | File Access Denied Due To Password Protection |
| 38 | I/O Attempt To Unopen File |
| 39 | Invalid Command Parameter |
| 40 | File Already In Directory |
| 41 | Attempt To Open File Already Open |

# Disk BASIC Functions

**CVD(*str*)** Converts to double-precision after GET.
```
A#=CVD(GRSPAY$)
```

**CVI(*str*)** Converts to integer after GET.
```
PRINT CVI(II$)
```

**CVS(*str*)** Converts to single-precision after GET.
```
FK=CVS(T$)
```

**EOF(*b*)** End-of-file detector for buffer *b*.
```
IF EOF(3) THEN CLOSE 3
```

**INSTR(*pos*, *mainstr*, *substr*)** Returns number which indicates the position of the main string where the substring begins. If substring not in main string, zero is returned. If *pos* is omitted, *pos* = 1.
```
PRINT INSTR(S$, "VA")      X=INSTR(S$,Q$)
Y1=INSTR(0, S$, Q)
```

**LOC(*n*)** Gets current record number.
```
PRINT LOC(1)
```

**LOF(*n*)** Determines number of last (highest-numbered) record in specified file.
```
Y=LOF(5)
```

**MKD$(*x*)** Makes double-precision number ready for disk write (random access).
```
LSET AVG$=MKD$(3000.00001)
```

**MKI$(*n*)** Makes integer number ready for disk write (random access).
```
LSET AVG$=MKI$(3000)      LSET Y$=MKI$(Y5)
```

**MKS$(*x*)** Makes single-precision number ready for disk write (random access).
```
LSET AVG$=MKS$(3000.1)      LSET M$=MKI$(N)
```

**USR*n*(*x*)** Calls any one of up to 10 machine-language subroutines, *n*=0-9. If *n* is omitted, zero is used. See DEFUSR*n*.
```
X=USR0(T1)      F=USR7(Y)
```

PRINTED IN U.S.A.
8759039 381-NP

# Disk BASIC Statements

**CLOSE** Closes all open file-buffers or specified buffer(s).
```
CLOSE     CLOSE 1,2,8     CLOSE N
```

**CMD "A"** Returns to TRSDOS on error.
```
CMD "A"
```

**CMD "B"** Enable/Disable (BREAK) key.
```
CMD "B","ON"     CMD "B","OFF"
```

**CMD "C"** Deletes program remarks (R) or spaces (S).
```
CMD "C",R     CMD "C",S     CMD "C"
```

**CMD "D"** Displays directory for specified drive.
```
CMD "D:1"
```

**CMD "E"** Displays previous TRSDOS error.
```
CMD "E"
```

**CMD "I", *command*** Executes a command to TRSDOS; may overwrite BASIC.
```
CMD "I","HELP"
```

**CMD "J"** Changes calendar date from *source* to *destination*. mm/dd/yy can be changed to ddd/yy – yy/ddd can be changed to mm/dd/yy
```
CMD"J","08/12/81",D$
CMD"J","-64/201",D$
```

**CMD "L", *routine*** Loads Z-80 routine or program file into RAM.
```
CMD "L",JOBFILE
```

**CMD "O", *x, array (start)*** Alphabetizes (sorts) contents of an array. *x* is the number of items to be sorted; *start* is where the sorting process begins.
```
CMD"O",50,A(1)
```

**CMD "P", *status*** Checks printer status. *status* is a string variable.
```
CMD "P",X$
```

**CMD "R"** Turns real-time clock display ON.
```
CMD "R"
```

**CMD "S"** Returns control to TRSDOS.
```
CMD"S"
```

**CMD "T"** Turns real-time clock display OFF.
```
CMD "T"
```

**CMD "X", *target*** Cross-references program lines and line numbers. *target* can be a reserved word, string, or string variable.
```
CMD "X",GOTO     CMD "X","PRINT"
```

**CMD "Z"** Simultaneous output to Printer and Display (dual routing).
```
CMD "Z","ON"     CMD"Z","OFF"
```

**DEFFN** Defines a user-created function.
```
DEF FNA$(X)=STRING$(X,45)
```

**DEFUSR*n*** Defines entry point for machine-language subroutine called by USR*n*. If *n* is omitted, zero is used.
```
DEFUSR=&H5500     DEFUSR4=&H7D7E
```

**FIELD** Organizes a random file buffer into fields.
```
FIELD 3,16 AS NM$,25 AS AD$
```

**GET *b, record number*** Gets specified or next record from a disk file (random access); stores it in buffer *b*.
```
GET 1     GET 1,25
```

**INPUT #*b*** Inputs data from buffer *b* (sequential access).
```
INPUT #1,A,B
```

**KILL** Deletes a disk file.
```
KILL "PRG/BAS"     KILL "FILE:1"
```

**LINE INPUT** Line inputs from keyboard; (ENTER) ends input.
```
LINE INPUT A$     LINE INPUT "ENTER YOUR
  NAME?"!N$
```

**LINE INPUT #** Line inputs from disk into specified buffer; carriage return, end-of-file, 255th character ends input.
```
LINE INPUT #1,A$
```

**LOAD** Loads program file from disk. R option causes program to run, leaving open files open.
```
LOAD "PROG/BAS"     LOAD "PROG:2",R
```

**LSET** Left-justifies data into a random access field.
```
LSET CITY$="DULUTH"
```

**MERGE** Merges disk program with resident program. Disk program must be in ASCII format.
```
MERGE "PR/BAS"
```

**MID$ (*old, pos, len*) = *repl*** Replaces one portion of a string with another. If length option is omitted, same number of characters in the old string will be changed as the number of characters in the replacement string.
```
MID$(A$,3,4)="USAFX"     MID$(A$,5)="01"
```

**NAME *newline, startline, increment*** Renumbers program line numbers. *newline* is the new number of the first line which is to be renumbered. If omitted, 10 is used. *startline* is the line number where renumbering is to begin. If omitted, entire program will be renumbered. *increment* is the increment between successive renumbered lines. If omitted, 10 is used.
```
NAME 100,10,100     NAME     NAME ,,5
```

**OPEN *mode, b, file, n*** Opens file; assigns mode (I = input, O = output, R = random, E = Output to end-of-file); assigns buffer number *b*, *file* specifies filename; *n* specific number of files.
```
OPEN "R",1,"CLIENTS/TXT"
```

**PRINT #*b*** Writes data to file-buffer *b* (sequential access).
```
PRINT #1,A
```

**PUT *b, record number*** Moves data from file-buffer *b* into the specified record (random access). If record number is omitted, current record number is used.
```
PUT 1,25     PUT 1     PUT C,N
```

**RSET** Right-justifies data into a random access field.
```
RSET CITY$="SPOKANE"
```

**RUN *program*** Loads and executes disk program. R option leaves open files open.
```
RUN "PROG/BAS"     RUN "PRG:1",R
```

**SAVE *filename*** Saves BASIC program on disk. A option causes file to be stored in ASCII format.
```
SAVE "FL1/BAS:3"     SAVE "PRT/TXT",A
```

# Disk BASIC Debug Monitor Commands

**D** Displays memory contents.
D ADDRESS? = *aaaa* where *aaaa* is a hexadecimal number.

**X** Half-screen display.

**S** Full-screen display.

**M** Modify RAM. M ADDRESS? = *aaaa* where *aaaa* is a hexadecimal number.

**R** Change Register contents.
R*aa*,*bbbb* (SPACEBAR) where *aa* is one of the register pairs AF, BC, DE, HC, PC and *bbbb* is a hexadecimal value.

**I** Single-step.

**C** Single-step executing call.

**U** Up-dates display.

**;** Increment the first location on a half-screen display by 16; on full-screen, by 256.

**–** Decrement the first location on a half-screen display by 16; on full-screen, by 256.

**J** Jump the transfer of control from one location to another.
J ADDRESS? = *aaaa*,*bbbb* where *aaaa* specifies the hexadecimal location where execution begins and *bbbb* specifies the hexadecimal location of the breakpoint.

**Q** Quits or exits from debug.

**F** Disk file utility which allows you to load disk file into memory and change it.

# Disk BASIC Error Codes

| | |
|---|---|
| 51 | Field Overflow |
| 52 | Internal Error |
| 53 | Bad File Number |
| 54 | File Not Found |
| 55 | Bad File Mode |
| 58 | Disk I/O Error |
| 62 | Disk Full |
| 63 | Input Past End |
| 64 | Bad Record Number |
| 65 | Bad File Name |
| 67 | Direct Statement in File |
| 68 | Too Many Files |
| 69 | Disk Write-Protect |
| 70 | File Access |

# Disk BASIC Abbreviations & Special Characters

| | |
|---|---|
| &H | Indicates following number is a hexadecimal constant. |
| &O | Indicates following number is an octal constant. |
| ↑ | Lists previous line. |
| ↓ | Lists next line. |
| → | Lists current line. |
| ← | Edit current line. |
| (SHIFT) ↑ | Lists first line. |
| (SHIFT) ↓ Z | Lists last line. |
| L*xx* | List line *xx*. |
| E*xx* | Edit line *xx*. |
| D*xx* | Delete line *xx*. |
| A*xxx, xxx* | Automatic line numbering beginning at line *xxx*, incrementing by *xxx*. |

# TRS-80® MODEL III DISK SYSTEM

## TRS-80® MODEL III MICRO-COMPUTER SYSTEM

# Start-Up

The entire system (Computer and peripherals) should be OFF and the disk drives empty.

1. Turn all peripherals ON.
2. Turn the Computer ON.
3. Insert a System diskette into Drive 0. Close the drive door.
4. Press the RESET button. Once the system is initialized, TRSDOS will load and take control.
5. To start Disk BASIC, type BASIC (ENTER)
6. When BASIC asks HOW MANY FILES? type in the number of concurrent files you need or press (ENTER) (three concurrent files).
7. Then BASIC will ask MEMORY SIZE? Answer by typing in a specific number or press (ENTER) to enter Disk BASIC.
8. The Disk BASIC start-up message will appear followed by the READY prompt. The Computer is now ready for use.

Radio Shack®
The biggest name in little computers™